

REMARKS

Claims 1-55 were examined. Applicant has amended claims 1 and 49. No claims are cancelled, or are newly presented. No new matter has been introduced.

Rejections under 35 U.S.C. §112

Claims 1-55 are rejected under §112, second paragraph, as being indefinite. Claim 1, line 10 recites the limitation “validating generated test cases”; line 5, it is unclear what “test cases” are being referred to; in line 11, it is unclear what “test cases” are being converted to “test scripts”. The examiner examined claim 1 as the Examiner anticipates the claims will be amended to obviate the §112 rejections. In Claim 1, the phrase “test cases” is considered by the Examiner in all locations (i.e., not “generated” nor “converted”. The examiner requests applicants amend the claim to particularly point out and consistently set forth the limitations “test cases”.

Claims 2-44 are rejected based on virtue of their dependency of the rejected claim 1.

Claim 49 recites the limitation “the logic using semantic” in line 8. There is insufficient antecedent basis for this limitation in the claim. The phrase is considered to read as “logic using semantic”.

Claims 50-55 stand rejected based on virtue of their dependency on the rejected claim 49.

The claims have been amended to overcome these grounds of rejection.

Rejections under 35 U.S.C. §102

Claims 1-55 stand rejected under §102(b) as anticipated by “WinRunner 7.0 Tutorial”, Mercury Interactive Corp. 2000 (WinRunner).

This ground of rejection is respectively traversed.

In one embodiment of the present invention, as set forth in claim 1, a method for generating test scripts uses semantic analysis to decompose test cases into applications, external interaction sequences and input data. Test cases are generated using rule based generation of test cases from an abstract representation that includes application states, external interaction sequences and input data of test cases from data

stores. Each application state is a set of application objects associated with a set of attributes and their values, or represents a runtime snapshot of an application under test which defines a context of external interaction. The rule based test cases are validated. The rule based test cases are converted to test scripts that are platform independent. The rule based test cases can be used to generate scripts in different target environments.

WinRunner creates tool dependency and reduces ability to share the tests. Additionally, WinRunner does not use semantic analysis to convert test cases to abstract representations, and test data is not separated from test steps in abstract representations.

WinRunner supports two modes of development of test scripts: manual script development and recording of user interactions. In WinRunner the tool records user interactions into a script form which can be played back to execute the tests automatically without manual intervention. One of the disadvantages of this approach is that the scripts generated through recording are too fragile and requires additional manual efforts to ensure reusability for parameterization. Another disadvantage is that the scripts created using one tool will not run on another tool. This reduces the ability to share the test scripts with customer or vendors if they use a different tool.

The present invention specifically addresses the disadvantages of the record and playback tool of WinRunner. With the present invention, the test scripts are converted to a platform independent format (abstract representation) and the test cases can be shared seamlessly. Test scripts can be rebuild from abstract representation for multiple platforms, and the test cases can be executed in any platform from any vendor.

WinRunner uses tool specific scripting as compared to the present invention that uses abstract representation. WinRunner has its own specific scripting language for recording and playback.

This is clearly distinguished from the present invention where test scripts are converted to an abstract representation during the import, and then the test scripts can be generated in multiple languages. we remove tool dependency and increases reuse.

With WinRunner, test scripts contain test data embedded in test steps. Parameterization is done manually to remove data dependency and enable scripts to run with multiple test data. With the present invention, test data is separted from test

steps in abstract representations. This makes it easy to run same test cases with multiple data sets.

With WinRunner, application states, external interaction sequences (test steps) and data are mixed together within the scripts. It is very difficult to reuse or modify the test cases later as a result. This is distinguished from the present invention which uses abstract representations to distinguish these entities.

CONCLUSION

It is submitted that the present application is in form for allowance, and such action is respectfully requested. The Commissioner is authorized to charge any additional fees which may be required, including petition fees and extension of time fees, to Deposit Account No. 08-1641 (Docket No. 07464-0004).

Respectfully submitted,
HELLER EHRMAN LLP

Date: 12-19-07


Paul Davis, Reg. No. 29,294

275 Middlefield Road
Menlo Park, CA 94205
(650) 324-7041
Customer No. 25213